COMP62421 Querying Data on the Web

Explore the effectiveness and efficiency of SPARQL query in DBpedia

Yazhuo Cao
Department of Computer Science
University of Manchester

Content

1. Introduction

In this report, a set of SPARQL queries will be written to explore how the information contained in the Mondial database is available in DBpedia and how effective and efficient the queries are under a limited time.

The following Mondial table is used for the bases of these queries:

- Country
- City
- Continent
- Organization
- Province

Queries are firstly written to explore these subjects and their predicate in DBpedia and secondly are to retrieve values related to these subjects corresponding to the attribute of the Mondial table.

2. Exploratory Queries

2.1 Country

Country is defined in Wikipedia, where DBpedia extracts its information from, as a distinct territorial body or political entity. According to Wikipedia, there are currently 233 countries on earth (*Lists of countries and territories*, no date). Mondial contains information about countries and similar areas of the world in its Country table which has 244 rows.
All information returned in Mondial are countries that currently exist, therefore any collapsed country in Wikipedia must be filtered out.

The information on DBpedia is dependent on Wikipedia and a program converter is used to extract and structure the information (dbpedia, no date), meaning it is highly dependent on how the wiki editor phrases information.
Both UK and Pitcairn island have iso code in their information card on Wikipedia, but only Pitcairn islands have the predicate iso code in DBpedia.

Explore query for Country	
Time spend	1 hour 25 minutes
Query experimented	Comment
SELECT DISTINCT ?C WHERE {?C a dbo:Country. ?C a dbo:PopulatedPlace.	• Returns 5615 results which are more than expected. • Includes subjects that are a location related

Explore query for Country	
ORDER BY (?C)	to a but is not itself, a country and subjects that are a time period of a country and country that no longer existed.
SELECT DISTINCT ?C WHERE {?C a dbo:Country. ?C a dbo:PopulatedPlace. FILTER NOT EXISTS { ?C dbp:dateEnd ?dateEnde }}	• Returns 4032 results • Many collapsed countries are filtered out however some remain as they do not have the predicate dbp:dateEnd connect to its subject.
SELECT DISTINCT ?C WHERE {?C a dbo:Country. ?C a dbo:PopulatedPlace. ?C a umbel-rc:Country. FILTER NOT EXISTS { ?C dbp:dateEnd ?dateEnde } }	• A new restraint on the object is added to the query using the type Country from another namespace umbel. • Some countries such as Syria, are not included in the result this time, suggesting not all existing countries include this object as their type therefore this can not be used to find existing countries.
SELECT DISTINCT ?C WHERE {?C a dbo:Country. ?C a dbo:PopulatedPlace. ?C a ?y. FILTER regex(?y, "yago/Country", "i") . FILTER NOT EXISTS { ?C dbp:dateEnd ?dateEnde }}	• A new object is added to replace the previous one to explore the country type using a different namespace. • 1663 results were returned, including Syria. • Some subjects such as summit are still included in this query which needs to be filtered out.
SELECT DISTINCT ?C WHERE {?C a dbo:Country. ?C a dbo:PopulatedPlace. ?C a ?y. FILTER regex(?y, "yago/Country", "i") . {?C dbp:isoCode ?code. FILTER NOT EXISTS { ?C dbp:dateEnd ?dateEnd1 } }	• Research in Wikipedia shows iso code is a standard defining code for the names of countries or a state, which is also the value of the code column in the Country table in Mondial, therefore it is used to filter the currently existing countries. • However, this query results in only 18 results, suggesting not all subjects in RDF that is a country have dbp:isocode as one of its predicates.
SELECT DISTINCT ?C WHERE {?C a dbo:Country. ?C a dbo:PopulatedPlace. ?C a ?y. FILTER regex(?y, "yago/Country", "i") . ?C dbo:countryCode ?phoneCode. FILTER NOT EXISTS { ?C dbp:dateEnd ?dateEnde }}	• Further research in Wikipedia shows countries also each have their country code for calling, therefore a new predicate is used to replace iso code in the query. • This returns 218 results, however, they are distinct from the 18 results previously retrieved, which is a problem are the 18 results are also in the Mondial database.

Explore query for Country	
SELECT DISTINCT ?C WHERE {?C a dbo:Country. ?C a dbo:PopulatedPlace. ?C a ?y. FILTER regex(?y, "yago/Country") . {?C dbp:isoCode ?code. FILTER NOT EXISTS { ?C dbp:dateEnd ?dateEnd1 } }UNION{ ?C dbo:countryCode ?phoneCode. FILTER NOT EXISTS { ?C dbp:dateEnd ?dateEnd2} }}	● This returns 236 results, a combination of the two queries above, making it closer to the number of results Mondial database returns for countries but the result only partial overlap.

The final query is fairly effective, it contains most of the countries that are available through the relational database Mondial. Mondial contains 244 countries and 236 are returned from this query.

It took over an hour to explore the information content for countries in the DBpedia and write the query using SPARQL. The problems encountered while exploring and writing the query are documented in the timesheet table above.

2.2 City

The term "city" refers to a permanent, large human settlement with administratively defined boundaries in Wikipedia (*City*, no date). The city table in Mondial contains information about cities without including a definition.

Explore query for City	
Time spend	1 hour 20 minutes
Query experimented	Comment
SELECT DISTINCT ?C WHERE {?C rdf:type dbo:City. }	● Returns objects that are related to a city but not a city themselves such as 2009–10 Stoke City F.C. season, Sociology of Manchester and roads such as N2_road_(Belgium).

Explore query for City	
SELECT DISTINCT ?C WHERE {?C rdf:type schema:City. }	● A city property of a different prefix is used this time, schema:city does not return the object in the example above. It does, however, return the City of London, which is both a city and a borough, that is not included in the Mondial database. This suggests the predicate does not have the same definition of City as Mondial. ● Both Manchester and Manchester City Region are returned in this query however the latter is not a city itself, meaning the results need further filtration.
SELECT DISTINCT ?C WHERE {?C rdf:type schema:City. ?C rdf:type dbo:City. ?C rdfs:label ?y FILTER regex(?y, "Man", "i") . } ORDER BY (?C)	● Using both objects does not filter out Manchester City Region. ● FILTER is added here to confirm Manchester City Region will be included in the result if it has both schema:City and dbo:city as objects to its rdf:type predicate as only 10,000 results can be returned and there might be more results fitting the query.
SELECT DISTINCT ?C WHERE {?C rdf:type schema:City. ?C rdf:type dbo:City. ?C a ?y. FILTER regex(?y, "yago/City", "i") . } ORDER BY (?C)	● A new object with a different namespace, yago, is added to the query. This filters out Manchester City Region. ● However, in the result, there is Vaishali (ancient city) which was a city but now is an archaeological site. It has no predicate such as endDate that can show the city is no longer populated which means it can not be filtered out from the result.

The final query is effective in the sense that it fathers all the cities that are available through Mondial, it contains more cities than what is available through the relational database Mondial. Mondial contains 3350 cities and over 10,000 are returned from this query.

External databases were consulted that there are more than 1000 cities in the world (*List of all city's in the world with Latitude and Longitude*, no date; dbpedia, no date), however, as the Mondial database only has 3350 rows in its table City, it shows Mondial has certain criteria for what is a city and what is not but this information is not shown in its documentation.

It took over an hour to explore the information content for cities in the DBpedia and write the query using SPARQL. Part of the problem was because the size of the result returned was too much larger than Mondial and some filtering was attempted before the external

databases were consulted. The problems encountered while exploring and writing the query are documented in the timesheet table above.

2.3 Continent

A continent is any of several large landmasses (*Continent*, no date), identities by convention rather than strict criteria which means one can say there are seven or six or five continents and they will be correct. Mondial contains five continents in its Continent table: Africa, America, Asia, Australia/Oceania, Europe.

Explore query for Continent	
Time spend	20 minutes
Query experimented	Comment
SELECT DISTINCT ?C WHERE {?C rdf:type dbo:Continent. }	● The starting point would be to first filter the object of type continent. There are multiple predicates of continent in DBpedia with different namespace but Dbo is the most commonly used one in DBpedia therefore dbo:Continent is used as a starting point of the query. ● 68 results were returned, including lost continent Mu(mythical) and freshwater fish of Australia.
SELECT DISTINCT ?C WHERE {?C rdf:type dbo:Continent. ?C rdf:type schema:Continent. }	● Using both objects in the query filters out the mythical continent and freshwater fish of Australia however still returns subjects such as Soviet Central Asia which is a section of Central Asia.
SELECT DISTINCT ?C WHERE {?C rdf:type dbo:Continent. ?C rdf:type schema:Continent. ?C rdf:type umbel-rc:Continent }	● Experimenting with a new namespace for continent returns 7 results that look reliable: Latin America and North America on top of the 5 results from the Mondial database. However, Latin America is not a continent but just a region, suggesting the umbel-rc:Continent object is not an accurate identification object. ● The link to the namespace umbel-rc is outdated and the current link in the website is for an online casino company.
SELECT DISTINCT ?C WHERE {?C rdf:type yago:WikicatContinents. ?C rdf:type dbo:Continent.	● This combination of objects returns 12 continents containing the 5 individual continents from Mondial and merged continents such as Eurasia, which is so far

Explore query for Continent	
?C rdf:type schema:Continent. }	the most correct set of objects returned.
SELECT DISTINCT ?C WHERE {?C rdf:type yago:WikicatContinents. }	● A new experiment using the yago object alone proves to be unsuccessful as the result includes other historical continents such as Pangaea, this shows a single type can not be used to find the ideal result.

The final query is effective. Mondial contains 5 continents and 12 are returned from this query. There is no strict definition of a continent which means both Mondial and the query returns the accurate result.

It took 20 minutes to explore the information content for continents in the DBpedia and write the query using SPARQL. This is a very short time of experimenting with different prefixes. At least one prefix used in DBpedia is now out of date. The prefix umbel-rc now links to an online casino website (umbel-rc), assuming it is because the domain expired and is brought by other people.

The problems encountered while exploring and writing the query are documented in the timesheet table above.

2.4 Organization

The term Organization is defined as an entity comprising one or more people and having a particular purpose in Wikipedia (*Organization*, no date). The Mondial table Organization, on the other hand, contains information about political and economical organizations. Therefore when the SPARQL query was constructed, it only aimed to return political and economical organizations, in line with the Mondial table definition.

However, even though according to the relational schema of Mondial, only political and economical organizations are present in its Organization table, it also contains the World Health Organization (*World Health Organization*, no date) . Some may argue that WHO is political, nevertheless it is neither political nor economical officially therefore it should not be part of the result returned from DBpedia.

Explore query for Organization	
Time spend	1 hour 5 minutes
Query experimented	Comment
SELECT DISTINCT ?C	● The starting point to explore the predicate for

Explore query for Organization	
WHERE {?C rdf:type dbo:Organisation. }	organisation, using the same definition as the Mondial database, is to loop for all possible organizations in DBpedia to find out their predicate.
SELECT DISTINCT ?C WHERE {?C rdf:type dbo:Organisation. ?C rdf:type yago:WikicatPoliticalOrganizations. }	● As yago namespace has shown to be rather effective in previous tables, it is added first into the query to explore political organization. ● 129 results were returned with results such as the Communist Party of the Soviet Union, a political organization that no longer exists. Further investigation shows this object has a predicate dbp:banned with the date the party is banned as its value.
SELECT DISTINCT ?C WHERE {?C rdf:type dbo:Organisation. ?C rdf:type yago:WikicatPoliticalOrganizations. FILTER NOT EXISTS { ?C dbp:banned ?x } FILTER NOT EXISTS { ?C dbo:dissolutionYear ?y } }	● Two FILTER is added to the query above to removed any object with the predicate dbp:banned or the predicate dissolutioniYear, both indicated the organisation no longer exist. ● This returns 118 results, without the Communist Party of the Soviet Union. ● However, another Soviet party, the Neo-Communist Party of the Soviet Union is still in the result. In its abstract, it explains the party only existed between 1974 and 1985, nevertheless as it has no predicate containing this information, therefore, it can not be filtered out from the result using these predicate.
SELECT DISTINCT ?C WHERE {?C rdf:type dbo:Organisation. ?C rdf:type yago:WikicatPoliticalOrganizations. FILTER NOT EXISTS { ?C rdf:type ?b. FILTER regex(?b,"banned", "I")} FILTER NOT EXISTS { ?C dbp:banned ?x } FILTER NOT EXISTS { ?C dbo:dissolutionYear ?y }}	● The Neo-Communist Party of the Soviet Union has no predicate that can indicate it no longer exist, however, it is of type yago:WikicatBannedCommunistParties, therefore a new filter is added to filter out any subject with predicate rdf:type and its object value contain the string “banned”. ● This filters out 3 results and 116 remained. ● The three FILTER NOT EXIST is separate to subject that is valid against any of the three graph patterns are filtered out, instead of valid again all three graph patterns.
SELECT DISTINCT ?C WHERE {?C rdf:type	● The rdf:type yago:WikicatInternationalEconomicOrganizati

<table>
<tr><th colspan="2">Explore query for Organization</th></tr>
<tr><td>dbo:Organisation.
{?C rdf:type
yago:WikicatInternationalPoliticalOrganizations.
FILTER NOT EXISTS {
?C rdf:type ?b.
FILTER regex(?b,"banned", "I") .}
FILTER NOT EXISTS { ?C
dbp:banned ?x }
FILTER NOT EXISTS { ?C
dbo:dissolutionYear ?y }
}UNION{
?C rdf:type
yago:WikicatInternationalEconomicOrganizations.
FILTER NOT EXISTS {
?C rdf:type ?b1.
FILTER regex(?b1,"banned", "I") .}
FILTER NOT EXISTS { ?C
dbp:banned ?x1 }
FILTER NOT EXISTS { ?C
dbo:dissolutionYear ?y1 }}}</td><td>ons. Is added so the query to return both political and economical organization.
● The two are connected using UNION.
● 220 results are returned containing organisations that are mostly existing currently. Not all of the 220 organisations are checked individually, only a sample test is conducted manually.</td></tr>
<tr><td>SELECT DISTINCT ?C
WHERE {?C rdf:type
dbo:Organisation.
{?C rdf:type
yago:WikicatInternationalPoliticalOrganizations.
FILTER NOT EXISTS {
?C rdf:type ?b.
FILTER regex(?b,"banned", "I") .}
FILTER NOT EXISTS {
?C rdf:type ?d.
FILTER
regex(?d,"WikicatFormerInternationalOrganizations", "I") .}
FILTER NOT EXISTS { ?C
dbp:banned ?x }
FILTER NOT EXISTS { ?C
dbo:dissolutionYear ?y }
}UNION{
?C rdf:type
yago:WikicatInternationalEconomicOrganizations.
FILTER NOT EXISTS {</td><td>● Follow up from the previous query, organization such as Union of African States has been disband but again has no predicate to indicate that therefore a new filter is added to remove any subject of type that contain the string WikicatFormerInternationalOrganizations. Because there are no schemas for DBpedia, the only way to make sure all results are correct is to go through every single result which is not possible under a limited time period like this.
● 121 results are returned.</td></tr>
</table>

Explore query for Organization	
?C rdf:type ?b1. FILTER regex(?b1,"banned", "I") .} FILTER NOT EXISTS { ?C rdf:type ?d1. FILTER regex(?d1,"WikicatFormerInternationalOrganizations", "I") .} FILTER NOT EXISTS { ?C dbp:banned ?x1 } FILTER NOT EXISTS { ?C dbo:dissolutionYear ?y1 }}}	

The knowledge base Yago where predicate yago is from combines Wikidata and schema.org (*YAGO: Getting Started*, no date), which could be why it is an ideal type to find the entity needed.

Mondial has 168 rows in its table for Organization and only 121 results are returned from the SPARQL query, meaning the query is not very efficient in gathering information that is available through Mondial. However as mentioned before the timesheet for Organization, Mondial contains WHO in its table even though it is neither a political nor economical organization, and the International Court of Justice does not handlel political matters and is not involved in the economical matter either. As the information in Mondial does not seems to be following its own definition that the SPARQL query is based on, it is hard to judge exactly how efficient the query is other than it is somewhat efficient.

A little over an hour was used to explore and write the queries above as shown in the timesheet. Specific comments are also included along with the queries.

2.5 Province

Province is the most difficult table to write an exploratory query for. Wikipedia says a province is almost always an administrative division within a country or state and different countries have different definitions and classifications of their provinces (*Province*, no date). The table Province in Mondial contains information about administrative divisions. However, according to the information in Mondial, the only province in Ireland is Ireland while Ireland currently has 26 administrative county councils (*Counties of Ireland*, no date).

Mondial does not have efficient information about how it classifies provinces and DBpedia relies on the information on Wikipedia and predicates from different namespaces. The lack of information and clarity on both makes it difficult to return the subjects looking for.

Explore query for Province	
Time spend	1 hour 10 minutes
Query experimented	Comment

SELECT DISTINCT ?C
WHERE {
?C a dbo:Location.
?X dbp:region ?C.
?C a ?y.
FILTER regex(?y, "Region", "i") .
}
	• There is no predicate dedicated to the province, therefore there is no clear way to filter out the province and more complicated paths have to be taken.
	• This result looks for all locations, subject types containing the string Region and subject that is a region of another subject.
	• Nevertheless, in the 5190 results returned there are many city regions that are not an administrative district.

SELECT DISTINCT ?C
WHERE {
?C a dbo:Location.
?X dbp:region ?C.
?C a ?y.
FILTER regex(?y,
"AdministrativeDistrict", "i") .
}
	• Instead of filtering for keyword “Region” in the subject types, AdministrativeDistrict is used instead as some provinces, such as North West England, is also of type yago:AdministrativeDistrict108491826.
	• The keyword change changed the number of results to 3939.
	• In these 3939 results, there are some subjects with “city” in the name but are also administrative territory.

SELECT DISTINCT ?C
WHERE {
?C a dbo:Location.
?X dbp:region ?C.
?C a ?y.
FILTER regex(?y,
"AdministrativeDistrict", "i") .
?C dbo:subdivision ?z.
?z a dbo:Country
}
	• A new condition is added to this query to only return a subject that is a subdivision of a country.
	• This query returns 197 results. It contains the city Birmingham from the United Kingdom. Birmingham is an administrative distributive however it is a subdivision of West Midland which is a province that is returned in Mondial.

More results are returned from the DBpedia using the SPARQL queries above. It does gather the information that is available through Mondail which shows the query is efficient however there is one thing worth pointing out.

Mondial does not have the complete, suitable result for the province which itself defines as administrative divisions. This definition is not, strictly speaking, the same as administrative district however for Ireland, the counties are historic administrative divisions of the island (*Counties of Ireland*, no date), and counties are administrative district according to DBpedia.

Mondial also returns Ireland as the one and only province in Ireland as mentioned before the timesheet, which is hard to judge but is most likely a mistake.

A little over an hour was used to explore and write the queries above as shown in the timesheet. Specific comments are also included along with the queries.

2.6 Conclusion from the exploratory queries

Unlike Mondial, DBpedia stores its data in RDF format and has no schemas. This creates a huge difficulty in terms of finding the subject/entity needed as there is no confirmation on which predicate would be available for which type of entity. For example, there is no single predicate, or a certain set of predicates combined that would return, and only return the cities, or other entity from the Mondial table that was explored earlier in this report.

3. Retrieval Queries

In this section of the report, queries will be written to try to extract information from the five tables explored above using the Mondial schema as a basis for the information that needs to be extracted.

3.1 Country

The information that needs to be retrieved from DBpedia is Name, Code, Capital, Province Area and Population.

Given the lack of schemas, there is no definitive universal predicate for all countries that contain the information required. For example, most countries only have country phone code and only some territorial has their iso code linked to their entity.

Retrieval query for Country	
Time spend	30 minutes
Query experimented	Comment

SELECT DISTINCT ?C ?name ?cap ?area ?pop
WHERE {?C a dbo:Country.
?C a dbo:PopulatedPlace.
?C a ?y.
FILTER regex(?y, "yago/Country") .
OPTIONAL{
?C foaf:name ?name.
?C dbo:capital ?cap.
?C dbo:populationTotal ?pop.
?C dbo:area ?area.
}{
?C dbp:isoCode ?code.
FILTER NOT EXISTS { ?C dbp:dateEnd ?dateEnd1 }
}UNION{
?C dbo:countryCode ?phoneCode.
FILTER NOT EXISTS { ?C dbp:dateEnd ?dateEnd2}}}
	• The information required are included in an OPTIONAL clause in the query to make sure any country that does not use those predicate for this information are still shown in the result, instead of removed from it.
	• Some country has multiple values under one predicate. This means they would occupy multiple rows on the result table and therefore cause redundancy in the result table.

This query is somewhat effective. Not every country gathered has all the information, probably is because of the difference in the predicate different country uses, which is why

the information is gathered using an optional clause, to show which country does not use the predicate included in the query.

The code in the Mondial table contains information about vehicle registration code which is not in the countries Wikipedia page and is not connected with any predicate in DBpedia which is why it is not included in the query. The closes value to it is the ISO 3166 code of each country but it is not a predicate in all countries either, as explored before in the exploratory query for the country in section 2.1.

3.2 City

The information that needs to be retrieved from DBpedia is Name, Country, Province, Population, Latitude, Longitude, Elevation.

One thing to note is that there is nothing but NULL under the elevation(distance above sea level) column in the Mondial table even though it is included in the Mondial schema, there is no valid information stored about it.

Some city has multiple pair of latitude and longitude information. Further investigation shows they have multiple geo coordinate values in their infobox on Wikipedia, for example, the coordinate value of the city and important landmarks in the city. DBpedia would classify them both as geo coordination of the city and connect them separately to the predicates, geo:lat and geo:long. This means when there is n pair of coordinates, it will be stored in n^2 rows in the result table and is a serious redundancy. For example, if Wikipedia has coordinate A1B1 and A2B2, there would be 4 rows in the table: A1B1, A1B2, A2B1, A2B1.

Retrieval query for City	
Time spend	30 minutes
Query experimented	Comment
SELECT DISTINCT ?C ?name ?iso ?pop ?prov ?ele ?lat ?long WHERE {?C rdf:type schema:City. ?C rdf:type dbo:City. ?C a ?y. FILTER regex(?y, "yago/City", "i") . OPTIONAL{ ?C dbp:name ?name. ?C dbo:isoCodeRegion ?iso. ?C dbo:populationUrban ?pop. ?C dbo:subdivision ?prov. ?C dbo:elevation ?ele. ?C geo:lat ?lat. ?C geo:long ?long.}}	• The information required is included in an OPTIONAL clause in the query to make sure any country that does not use those predicates for this information are still shown in the result, instead of removed from it. • Some city has multiple values for predicate name, province, and even latitude and longitude. This makes the table includes a lot of redundant data.

The query is fairly efficient and actually return some value for the elevation of each city, unlink Mondial which returns NULL. Many cities do not have a predicate for the country they belong to and ISO code that contains its country's iso code is more common and is closer in value to the vehicle registration code that table City use as a foreign key from table Country. There is also no predicate for the total population from the city, many of them separate it as population metro and population urban.

About 30 minutes was spent on writing and exploring that predicate is included in the pages for the city. Comments are written in the timesheet.

3.3 Continent

Continent table in Mondial only has two columns: Name and Area.

Retrieval query for Continent	
Time spend	5 minutes
Query experimented	Comment
SELECT DISTINCT ?C ?area WHERE {?C rdf:type yago:WikicatContinents. ?C rdf:type dbo:Continent. ?C rdf:type schema:Continent. ?C dbo:areaTotal ?area }	● This is, so far, the easiest table to retrieve information for. Not just because there are only two columns, but also because the information on the continents is more complete than in some smaller cities. ● One part to note is that the continents have no name predicate but the name is stored in the label predicate. ● The Indian Subcontinent does not have the same predicate as the other continents, however, despite the name, it is also not an actual continent therefore if the OPTIONAL clauses are removed, only continents will be returned.

The query is efficient. Retrieving the two pieces of information, Name and Area, also filters out the Indian subcontinent. The returned 11 results include the 5 continents in the Mondial database, meaning the query is more efficient however as the Continent in Mondial is also used as a foreign key in other tables, therefore, the continents should not overlap each other in the area, whereas in DBpedia it is more about store the information in structure which means continent and merged continent, for example, Asia and Eurasia, can co-exist without an issue.

About 5 minutes was spent on writing and exploring that predicate is included in the pages for the continent which is very short as there are only 2 columns. Comments are written in the timesheet.

3.4 Organization

Continent table in Mondial has the following information: Abbreviation, Name, City, Country, Province, Established.

Retrieval query for Organization	
Time spend	30 minutes
Query experimented	Comment
SELECT DISTINCT ?C ?year ?label ?name ?abb ?country ?country1 ?city ?city1 WHERE {?C rdf:type dbo:Organisation. OPTIONAL{ ?C dbo:abbreviation ?abb. ?C dbo:foundingYear ?year. ?C rdfs:label ?label. ?C dbp:name ?name. ?C dbo:location ?country. ?country a dbo:country. ?C dbo:location ?city. ?city a dbo:city. ?C dbo:location ?prov. ?city a ?y. FILTER regex(?y, "AdministrativeDistrict", "i") . ?C dbo:headquarter ?country1. ?country1 a dbo:country. ?C dbo:headquarter ?city1. ?city1 a dbo:city. ?C dbo:headquarter ?prov1. ?prov1 a ?y1. FILTER regex(?y1, "AdministrativeDistrict", "i") .} —The rest of the query would be the same as the last explore query in the table *Explore query for Organization*, omitted here due to the length—	● Organizations does not share the same predicate like Mondial for its information. ● As seen in this query, both label and name need to be returned because some organization has the predicate for name and some do not, therefore another path is needed to find its name. ● The situation is the same with location and headquater. ● Mondial separated the location of the organization into CIty, Country and Province. This is also doable for DBpedia for City, Country and with some result for Province.

The query is should be fairly efficient however takes a long time to run. The location is attempted to be split into city, country and province, multiple predicates are also used for the name.
About 30 minutes were spent on exploring and writing the query, with comments written above and in the timesheet.

3.5 Provinces

Provinces table in Mondial has the following information: Name, Country, Population, Area, Capital, CapProv.

Retrieving the name of the capital and the name of the province where the capital belongs to would be difficult in DBpedia. DIfferent country has a different definition for the province as mentioned above in section 2.5, which means the province in some countries would not have a capital but sometimes only has the largest city in the province which Mondial classify as the capital of the province in its Province table. For example, North West England is a region with no administration capital but its largest city is Manchester (*North West England*, no date), which is shown as the capital of North West. This shows the information in Mondial is not accurate.

Retrieval query for Provinces	
Time spend	20 minutes
Query experimented	Comment

SELECT DISTINCT ?C ?z ?pop ?area
WHERE {
?C a dbo:Location.
?X dbp:region ?C.
?C a ?y. FILTER regex(?y,
"AdministrativeDistrict", "i") .
?C dbo:subdivision ?z.
?z a dbo:Country.
OPTIONAL{
?C dbo:populationTotal ?pop.
?C dbo:areaTotal ?area.
}}
	● As mentioned, the province has a different definition in different countries and therefore different attributes. Even the provinces of the same country do not guarantee they share the same predicate, making retrieving information more difficult and complex.
	● Retrieving capital and the province the capital belongs to is more complex.

The query is fairly efficient if the difference in the province in Mondial and province returned from section 2.5 is ignored. The country predicate returns the name of the country instead of the vehicle registration code which is a uniform theme throughout the entities returned from DBpedia.

About 20 minutes were spent on exploring and writing the query, with comments written above and in the timesheet.

3.6 Conclusion from the retrieval queries

DBpedia extracts information from Wikipedia using ita extraction framework that is currently mature against the semi-structured content like infoboxes (DBpedia, no date), however, the unstructured content extractor is less accurate.

Some of the retrieval queries above are very long because of the difference in predicate for the information that needs to be retrieved.

4. References

City (no date). Available at: https://en.wikipedia.org/wiki/City (Accessed: 17 December 2021).

Counties of Ireland (no date). Available at: https://en.wikipedia.org/wiki/Counties_of_Ireland (Accessed: 17 December 2021).

dbpedia (no date) *GitHub - dbpedia/fact-extractor: Fact Extraction from Wikipedia Text*. Available at: https://github.com/dbpedia/fact-extractor (Accessed: 17 December 2021).

YAGO: Getting Started (no date). Available at: https://yago-knowledge.org/getting-started (Accessed: 17 December 2021).

List of all city's in the world with Latitude and Longitude (no date). Available at: https://geokeo.com/database/city/ (Accessed: 17 December 2021).

Lists of countries and territories (no date). Available at: https://en.wikipedia.org/wiki/Lists_of_countries_and_territories (Accessed: 17 December 2021).

North West England (no date). Available at: https://en.wikipedia.org/wiki/North_West_England (Accessed: 17 December 2021).

Province (no date). Available at: https://en.wikipedia.org/wiki/Province (Accessed: 17 December 2021).

Continent (no date). Available at: Wikipedia, https://en.wikipedia.org/wiki/Continent.

Organization (no date). Available at: ttps://en.wikipedia.org/wiki/Organization.

World Health Organization (no date). Available at: https://en.wikipedia.org/wiki/World_Health_Organization (Accessed: 17 December 2021).